AMENDMENTS TO THE SPECIFICATION

Page 1, lines 3-19, replace as follows:

Field of the Invention

The present invention relates to a scheduler for packet switches, and more specifically, to a method for scheduling data packets from a plurality of input ports to at least one output port comprising the steps of storing data packets in a plurality of virtual output queues, a virtual output queue being arranged to store data packets from one of the plurality of input ports destined for a specific one of the at least one output port, and scheduling the plurality of virtual output queues.

Description of the Related Art

Scheduling packets in terabit switches and gigabit passive optical networks (PONs) requires a considerable amount of computation power. When a priority mechanism has to be deployed to manage traffic with different quality of service (QoS), the problem becomes even more complex. This complexity can be expressed as the total number of input queues in the system that needs to be scheduled per output port, i.e. the product of the number of input ports and the number of service classes. An algorithm is required to schedule packets of a large number of queues, according to their specific priority. The algorithm must be implemented efficiently in state-of-the-art technology, i.e. ASICs OR FPGAs.

Page 2, lines 3-18, replace as follows:

The article 'A Contention-Free Packet Scheduling Scheme for Provision of Quality-of-Service in Tbit/sec WDM Networks' by I. Elhanany, J. Nir. D. Sadot,, Optical Networks Magazine, July 2000, discloses a scheduling scheme for packet switches. An algorithm has been proposed

that claims in the order of N^{*2} log(N) operations per packet slot period, in which N is the number of output ports or destinations (the article relates to an N x N switch). This method employs sequential assertion of the different input ports using a round robin procedure, including a prioritised-matching scheme per input port to comply with diverse quality of service requirements. For large number of queues this is still to slow. It also does not address PON's.

Summary of the Invention

The present invention seeks to provide a scheduler for packet switches and PON's which is able to schedule packets of a large number of queues, according to their specific priority. The number of queues is equal to the number of input ports, or equal to the product of number of input ports and number of service classes (or priority classes) in the case of management of data traffic with different quality-of-service requirements.

Page 5, lines 11-20 replace as follows:

In further embodiment of the present scheduling system, the scheduling system comprises a plurality of scheduling trees associated with a first and a second output port. The scheduling system is arranged to activate the scheduling tree associated with the second output port if the first port is unavailable for the associated virtual output queue. This embodiment allows connectionless scheduling or protection switching, and can be implemented easily using additional logic circuitry.

Brief Description of the Drawings

The present invention will be clarified in more detail using a number

of exemplary embodiments, with reference to the accompanying drawings in which:

Page 6, lines 3- 21, replace as follows:

Fig. 7 shows a schematic diagram of a further implementation of the present invention in a passive optical network scheduler.

Detailed Description

Fig.1a shows a schematic diagram of an application of a scheduler for switching input data streams to specific output data streams. A switch 2 connects N_I input ports $1_1 \ldots 1_I$ to N_O output ports $3_1 \ldots 3_O$. Each of the input ports $1_1 \ldots 1_I$ can be connected to any of the output ports $3_1 \ldots 3_O$ by the switch 2. In order to avoid collision of data from different input ports $1_1 \ldots 1_I$, a schedule function is needed to regulate access to each of the output ports $3_1 \ldots 3_O$. A similar function is necessary in a passive optical network (PON) which is shown in Fig. 1b, where next to the switch 2, the input ports $1_1 \ldots 1_I$ are connected to N_G shared media $4_1 \ldots 4_G$. In the case of a PON, the schedule function not only has to regulate access to the output ports $3_1 \ldots 3_O$, but also access to the shared media $4_1 \ldots 4_G$. In the above mentioned cases, the switch 2 is considered to be single-stage non-blocking, i.e. the input ports $1_1 \ldots 1_I$ are directly connected to the scheduled output port $3_1 \ldots 3_O$. For the case of a PON, two separate schedulers may be implemented, one for the access to the shared media $4_1 \ldots 4_G$ and one for the access to the output ports $3_1 \ldots 3_O$. However, this requires additional input queues at the input of the switch 2 and reduces the end-to-end performance.